database available to the server, such as Microsoft's Access **131***a*. Thus small databases which have not yet invested in being able to gather information from an intranet resource, can use their own direct resources, and also be interrogated by the Web browser(s) **130**, or another web browser **136**. Remember that browser's **130** can also communicate with the Web server **131'** across the Internet, just as can a Web browser **136** located on the intranet **140** inside of the firewall illustrated by the intranet **140** dashed line shown in FIG. **11**. With a browser **136** in place at the Web Server **131'** location, that browser **136** can make requests, if authorized across the intranet to the Web Server **131** which can then utilize the DIS capsules provided by the DIS Server **133**.

Physically, the network **132** will have its own access server **135** preferably in the form of a TCP/IP server **135** to make the physical connection across the Internet. We illustrate in FIG. **11** this other logical layer as located on the network. This TCP/IP server supports the physical connections which are needed by the other logical higher levels of service supported on the network. The use of an InterNetwork Routing Protocol (INR) allows the logical coupling illustrated between a application processing server **134** to an external intranet application processing server **134'**. On each network there can be one or more web servers. A Hypertext document request asking for a field to be searched, as by a Hyperlink, could index to a server directly, e.g. a second web server **134"** on the same network which would have its own control program agent function duplicating the control program agent resident in web server **134**. Thus at the request homepage a menu which say if "Art&Literature search", when selected in a Hypelink setting, would index to a particular web server and a particular document within that web server's environment. This web server **134"** besides being linked to its own application processing server **133"** has a direct link, in the environment illustrated, to an MVS CICS, a transaction processing server for handling transaction processing. Such a solution allows CICS transaction processing to utilize the Internet to save transmission costs and still be located beneath a firewall for retention of data integrity. The outputs provided by the web server to the requested destination can be outside of the firewall, and in the form of results illustrated by the possible examples shown in FIGS. **3**, **5** and **8**.

While we have described our preferred embodiments of our invention, it will be understood that those skilled in the art, both now and in the future, may make various improvements and enhancements which fall within the scope of the claims which follow. These claims should be construed to maintain the proper protection for the invention first disclosed.

What is claimed is:

1. A method of fulfilling requests of a web browser client, comprising:

displaying an HTML document to said web browser;

invoking a control program agent;

receiving data entered by the user from the HTML document and passing said user entered data to said control program agent upon invocation as input parameters to said control program agent that were returned from said HTML document;

using an API set for invoking executable command files accessible via an associated command file object, provided as user entered data for supplying to said control program agent values selected by a user to control the processes of fulfilling a Web request and for determining the nature of a resultant fulfillment of the request from information resources supplied from the World Wide Web.

2. A method of fulfilling requests according to claim **1** wherein

said input parameters contain path environment data from the HTML document that referred to said control program agent.

3. A method of fulfilling requests according to claim **1** wherein said step of receiving data entered by the user from the HTML document and passing said user entered data to said control program agent upon invocation as input parameters to said control program agent that were returned from said HTML document includes environment variables comprising an identification of a command file to call, the name of the file containing the HTML statements to use when building an HTML document that reports results requested by the request, and the type of report file that is requested, as well as query data from the HTML document that referred to said control program agent.

4. A method of fulfilling requests according to claim **1** wherein said step of passing data from the HTML document that referred to said control program agent, includes passing variable information data which contains values selected by the user and/or default values selected by a HTML document designer set in said command file prior to its execution, all of which variable information data is stored in a buffer environment, and accessible by said command file and/or said control program agent and used by said control program agent to set variables for said command file.

5. A method of fulfilling requests according to claim **4** wherein the steps further comprise:

with the variable information now stored in a buffer, retrieving, with the control program agent from a store, command files and environment variables comprising an identification of a command file to call, the name of the file containing the HTML statements to use when building an HTML document that reports results requested by the request, and the type of report file that is requested, as well as query data from the HTML document that referred to said control program agent which a request has specified as associated with each command file for the Web request and loading from said buffer environment into memory associated with the control program agent the command file names available and the environment variables associated with each command file.

6. A method of fulfilling requests according to claim **5** wherein the steps further comprise: subsequent to the step of claim **5** initiating with the control program agent a connection between said control program agent and said API.

7. A method of fulfilling requests according to claim **6** wherein the steps further comprise after the performance of the step of claim **6**:

if required for control by a decision support system environment for said command file, logging onto a port or desktop for the assigned user by said control program agent.

8. A method of fulfilling requests according to claim **7** wherein the steps further comprise: once a command file is loaded into memory for said control program agent, the control program agent retrieves from its memory the command file environment variables associated with a command file passed to the control program agent once the control program agent is invoked.

9. A method of fulfilling requests according to claim **8** wherein the steps further comprise: subsequently, creating with said control program agent a data array stored in the control program memory containing the command file environment variables and said variable information data for